AIRE Findings Documentation

# Sphinx implementation

## Status

Success

## Instructions

Implement the provided AIRE documentation into a sphinx file structure.

## Findings

After some initial confusion about how to set up the documentation, the process was simple. I added the documentation to the preexisting basic sphinx project and most things integrated easily.

Text

Description automatically generated

The first major problem was that Sphinx does not natively read MarkDown files.

Text

Description automatically generated

To fix this, I needed to import the myst\_parser extension as well as add the .md and .html suffix’s to the config.py file.

The next issue was adding all the files to the toctree. At first, I simply tried to add the document that where the next level down and then add a new toctree to each .md for the files relevant to that individual file. This however kept causing issues as despite that sphinx can use .md files, .md file doesn’t use .rst file structure which mean I couldn’t add a .rst toctree to the file.

To temporarily fix this, I added orphan:true to the files to stop the errors and allow testing.

Graphical user interface

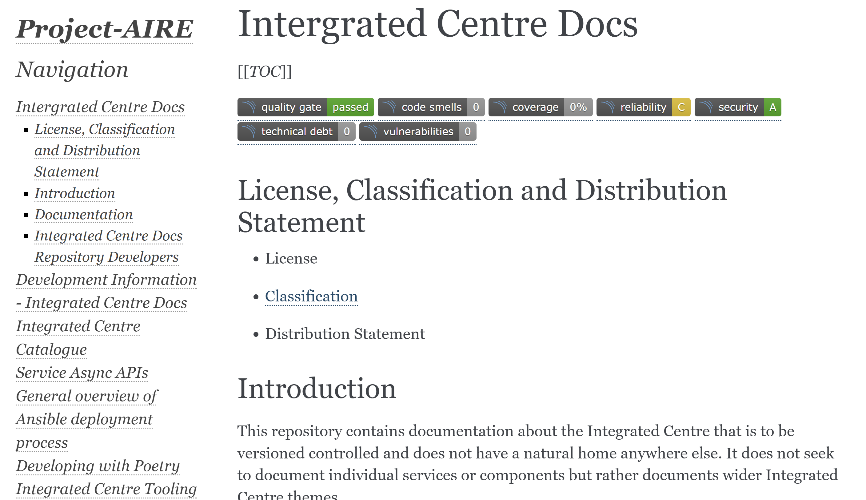
Description automatically generated with medium confidence

Unfortunately, the end result was what I originally thought. .rst toctree’s and .md TOC would not connect. The end result was just adding all the files to the original toctree.

Text

Description automatically generated

The end result worked, if not a bit wordy in the table of contents.



# Latex to PDF

## Status

Mostly successful (1 and 3 work. 2 works, but not to Lance’s preference.)

## Instructions

Use sphinx to convert Markdown to PDF with the following requirements:

1. Maintain relative links to sections within the PDF
2. Support headers and footers to allow classification banners when on Defence networks
3. Create one PDF file for N markdown pages including links between markdown pages.

## Findings

Latex is a built in Sphinx extension that allows the user to turn the entire project into a single pdf document (requirement 3).

The main errors that appear, seem to be dead links and .svg images. The dead links are not something I can fix so I simply commented them out.

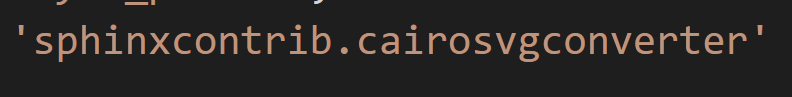
Text

Description automatically generated

Graphical user interface, text

Description automatically generated

I was finally able to get the images working using cairosvgconverter. However before that, I had to install MikTex Console (<https://miktex.org/>) as it was the primary tool used to install all additional requirements.



Graphical user interface, text, application, email

Description automatically generated

While I was able to get the images in, some are still bunch up a tiny bit and the status images are all duplicated. This has been deemed a minor issue.

While trying to fix the image issue, the links within the document disappeared as well as the table of contents. While I do not know exactly what caused this, they have since returned and are in working order (Requirement 1).

The final requirement was “Support headers and footers to allow classification banners when on Defence networks”.

While LaTeX natively allows minor headers and footers (project name and page number, etc) you cannot dynamically change a variable depending on the contents of another file (CLASSIFICATION.md). As such I have done some research and found “fancyhdr” (https://ctan.org/pkg/fancyhdr?lang=en) a recommended addon to LaTeX.

Text

Description automatically generated

So far I have managed to create a basic header that has the document details, page number, and the text “CLASSIFIED” on the PDF, however it is looking complicated to populate the field from a variable.

After way to many hours struggling to get the classification to be populated by the CLASSIFICATION.md, I have determined it cannot be done.

Text

Description automatically generated

I kept coming back to the same error “! Package catchfile Error: File ‘<name>’ not found”. At for I assumed it was a reference error, so I created a test file “text.txt” and placed it into the same location as config.py to minimize errors. Even this didn’t work. I tried many different techniques all centered around “catchfile” (as this was the technique continually recommended by most sources) and they all failed.

For the time being, the classification will need to be hard coded into the headers.

# DocBook to PDF

Status

Failed

Instructions

Use <https://github.com/msmid/markdown2docbook> to convert markdown documents of AIRE project into DocBook compatible files that will then be converted to PDF.

Findings

Text

Description automatically generated

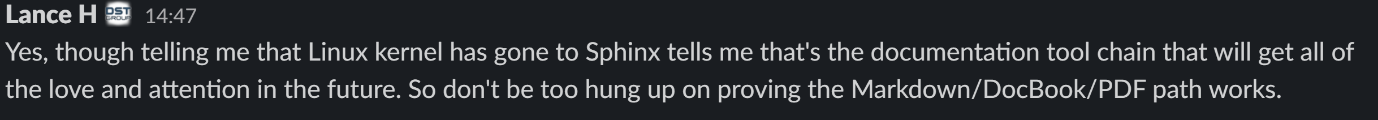
The github repository uses provided style sheets to convert the .md documentation into a DocBook compatible version. As we do not have a style sheet, I have just imported the 3 provided ones into the Sphinx directory and tried to connect them with the config.py file.

It has been made know to me that Docbook is not a work along WITH sphinx, but an alternative to. This means that the Markdown to DocBook test needs to be separated into a different folder. This also explains why I saw so little Docbook AND Sphinx data during research but lots of Docbook OR Sphinx. It also appear Sphinx is the preference for Linux Kernal and other sites so it seems it was changed for a reason (better functionality?).

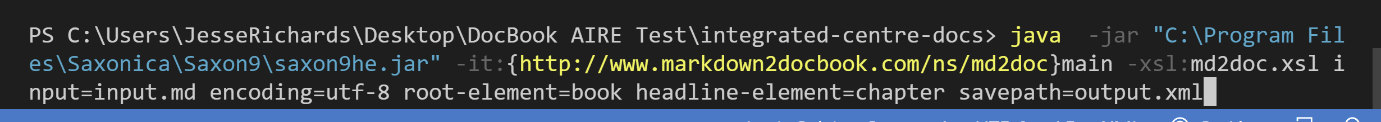
Timeline

Description automatically generated with medium confidence

Lance has also told me that if the general consensus is that Sphinx if better, than not to worry to much on Markdown to PDF. I will still try for a little but just to see if it even works.



After understanding the specifications of the task, I tried following the instructions laid out in the github README. Even after doing some editing to make the command relevant to my machine, it still produced an error.



Text

Description automatically generated

The error was referenced as if It was a pat of the Saxon addition, however the “main” template was referenced in the github README.

Text

Description automatically generated

Considering time and the overall importance of this aspect of research, I decided to drop Markdown to DocBook for the time being.

# PanDoc to Confluence